

Inserting and Updating in a Single Query Statement

Posted At : June 23, 2006 10:20 AM | Posted By : Mark Kruger

Related Categories: MS SQL Server, Coldfusion & Databases

This is a tip about something I do regularly. No, I'm not talking about forgetting to put down the seat. I'm talking about coding. Let's say you have an application that collects user data in a wizard-like format using a few steps. You allow folks to go back and forth between steps and edit what they've done. How do you handle the insert on that first step without creating duplicates when users return to that step? There are several approaches to this problem:

You could store something in the session or in a cookie to indicate the insert has already been done. You could *not* do an insert after step 1 and simply hold all the form variables till the "finish" button is pressed on the last step. Or... you could do something like this.

```
<cfquery name="put" datasource="myDsn">
  IF NOT EXISTS
    (SELECT * FROM myTable
     WHERE  email = '#form.email#'
     AND ....)
  BEGIN
    INSERT INTO myTable
      (email, fnmae, lname, ...)
    VALUES
      ('#form.email#',
       '#form.fname#',
       '#form.lname#',
       ...)
  END
ELSE
  BEGIN
    UPDATE  myTable
    SET     fname = '#form.fname#',
           lname = '#form.lname#',
    WHERE  email = '#form.Email#'

  END
</cfquery>
```

This SQL code checks to see if the user email (assumed to be unique in this example) exists. If it does *not* exist the statement does an insert. If it *does* exist the statement does an update (and yes I know if the user changed his email all bets are off).

3 things of note

I did not use <cfqueryparam ...> simply because of space and readability (for this blog). Normally all the form elements would be bound.

Secondly, this query doesn't return anything. If page 2 of your wizard needs an ID from the insert that could be a problem. So you will need to add @@identity or SCOPE_IDENTITY() to the query to get back the ID from the insert or a regular select to get the ID back from the update. That's pretty easy from within our context as well.

```
<cfquery name="put" datasource="myDsn">
  SET NOCOUNT ON
```

```
IF NOT EXISTS
    (SELECT * FROM myTable
      WHERE email = '#form.email#'
      AND ....)
BEGIN
    INSERT INTO myTable
        (email, fnmae, lname, ...)
    VALUES
        ('#form.email#',
         '#form.fname#',
         '#form.lname#',
         ...)
    SELECT SCOPE_IDENTITY() AS newId
END
ELSE
BEGIN
    UPDATE myTable
    SET     fname = '#form.fname#',
           lname = '#form.lname#',
    WHERE  email = '#form.Email#'

    SELECT id AS newid
    FROM   myTable
    WHERE  email = '#form.email#'

END
SET NOCOUNT OFF
</cfquery>
<!-- set a variable to the id returned --->
<cfset usersId = put.newid>
```

Finally, this code is specific to MS SQL server. There are ways to do this on other platforms. Perhaps some readers who are mySQL or Oracle gurus can add to the post.